The Dynamics of a Triple Celestial Body System

Taylor Larrechea

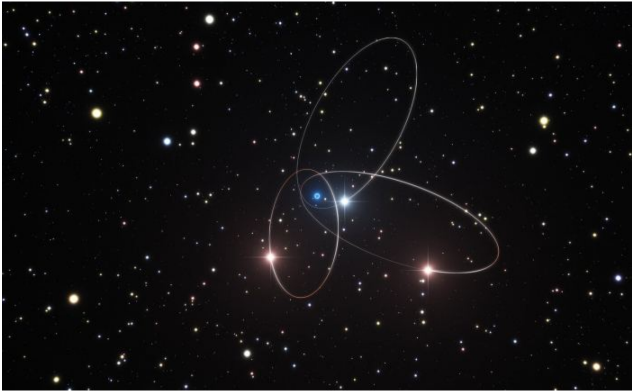
Colorado Mesa University

October 31, 2019

Table of Contents

Main Purpose of Project

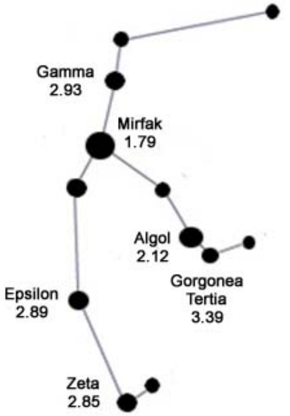
To plot the motion of three massive bodies in space orbiting one another using computational methods and Newtonian Gravitation.



Find initial conditions that would be optimal to use in the search of ternary star systems.

Multiple Star System

Most star systems are binary at the least. In fact, 85% of star systems are binary at least [3].



Methodology of Plotting Motion

We must first identify the differential equations that we plan to solve in order to plot the motion of our massive bodies in space

$$F(\vec{r}) = m\frac{d^2\vec{r}}{dt^2}$$

where $F(\vec{r})$ is

$$\vec{F} = \frac{GMm}{r^2}(\hat{n}). \tag{1}$$

Numerical methods are needed to solve these difficult differential equations.

Forward Euler Difference Scheme (1)

The differential equations that describe the acceleration of three bodies in space will have to be solved via numerical techniques with Python.

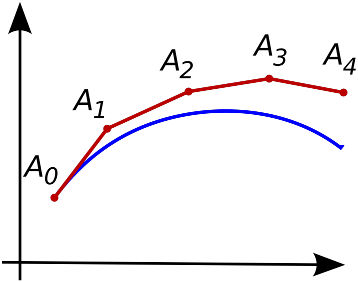
A typical method that is used to solve differential equations numerically is a Forward Euler Difference Scheme, or F.E.D.S for short [4]

$$y_{n+1} = y_n + hf(x_n, y_n). \tag{2}$$

A time step of 0.0001 seconds was used in this algorithm.

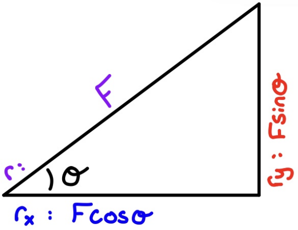
Forward Euler Difference Scheme (2)

Typically, the F.E.D.S is the simplest Runge-Kutta scheme there is for solving first order differential equations.



Forward Euler Difference Scheme (3)

To have an understanding of how our equations of emotion develop, examine the following image.



$$F = \frac{GMm}{r^2}$$

$$r = \sqrt{r_x^2 + r_y^2 + r_z^2}$$

$$F_x = \frac{GMm}{r^2}\cos\theta = \frac{GMm}{r^2}\frac{r_x}{r} = \frac{GMm}{r^3} r_x$$

$$F_y = \frac{GMm}{r^2}\sin\theta = \frac{GMm}{r^2}\frac{r_y}{r} = \frac{GMm}{r^3} r_y$$

Forward Euler Difference Scheme (4)

For simplicity's sake, we will only examine the two body equations.

Beginning with the distance between bodies

$$r_x = x_2 - x_1,$$

$$r_y = y_2 - y_1,$$

$$r_z = z_2 - z_1,$$

$$r = \sqrt{r_x^2 + r_y^2 + r_z^2}$$

we now have a way to quantify the distance between the two bodies in space.

Forward Euler Difference Scheme (5)

Now the force between the bodies of mass can be calculated

$$f_x = -\frac{Gm_1 m_2}{(r_x^2 + r_y^2 + r_z^2)^{3/2}}(r_x),$$

$$f_y = -\frac{Gm_1 m_2}{(r_x^2 + r_y^2 + r_z^2)^{3/2}}(r_y),$$

$$f_z = -\frac{Gm_1 m_2}{(r_x^2 + r_y^2 + r_z^2)^{3/2}}(r_z)$$

giving us three equations that we can use to find the force between our bodies of mass.

Forward Euler Difference Scheme (6)

Next with updating the velocities for mass one

$$v_{x_1} += -\frac{f_x}{m_1} \cdot dt,$$

$$v_{y_1} += -\frac{f_y}{m_1} \cdot dt,$$

$$v_{z_1} += -\frac{f_z}{m_1} \cdot dt,$$

we can now update the velocities in each direction of one body.

Forward Euler Difference Scheme (7)

We now can begin to update the positions of our bodies of mass. Updating the positions for mass one

$$
\begin{aligned}
x_1 &+= v_{x_1} \cdot dt, \\
y_1 &+= v_{y_1} \cdot dt, \\
z_1 &+= v_{z_1} \cdot dt, \\
t &+= dt,
\end{aligned}
$$

we are now in position to plot these positions.

Forward Euler Difference Scheme (8)

Let's examine the distance on mass 1 due to mass 2 and mass 3. The distances between the masses are

$$r_{12} = (r_{12x}^2 + r_{12y}^2 + r_{12z}^2)^{3/2},$$

$$r_{13} = (r_{13x}^2 + r_{13y}^2 + r_{13z}^2)^{3/2}.$$

With the above equations we can now calculate the force between the three bodies.

Forward Euler Difference Scheme (9)

The forces in each direction can now be calculated. The equations that were fed into python are

$$f_x = -\frac{Gm_1 m_2}{r_{12}}(r_{12_x}) - \frac{Gm_1 m_3}{r_{13}}(r_{13x}),$$

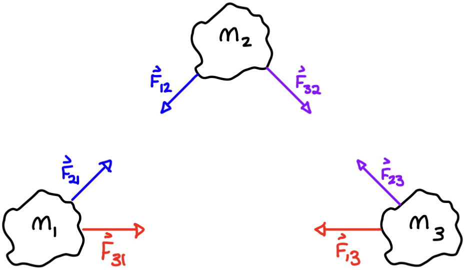
$$f_y = -\frac{Gm_1 m_2}{r_{12}}(r_{12_y}) - \frac{Gm_1 m_3}{r_{13}}(r_{13y}),$$

$$f_z = -\frac{Gm_1 m_2}{r_{12}}(r_{12_z}) - \frac{Gm_1 m_3}{r_{13}}(r_{13z}),$$

which now can be used to calculate new velocities and positions.

Forward Euler Difference Scheme (10)

The only difference between the two body and three body system is how the distance between the bodies are calculated and the forces between the bodies are calculated.

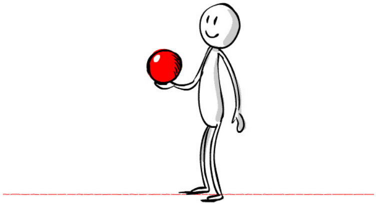


Forward Euler Difference Scheme (11)

Now that the groundwork is laid for how to solve the equations of motion, we can begin to start on the actual work that involves solving these differential equations.

Simple 1-D Motion (1)

The simplest scenario for this project is 1-D motion. First we will solve for the motion of a ball being thrown in the air so high that we cannot just assume $\vec{a} = g$, assuming that it only has vertical motion.



Simple 1-D Motion (2)

We will first start with Newtonian Gravity

$$\vec{F} = \frac{GMm}{r^2} \tag{3}$$

where r is the distance from the center of a body of mass. Using the formula for Force where mass is constant we have

$$\vec{F} = m\vec{a}. \tag{4}$$

Rearranging equations (3) and (4) we have our equation for acceleration

$$\vec{a} = \frac{GM}{r^2}. \tag{5}$$

Simple 1-D Motion (3)

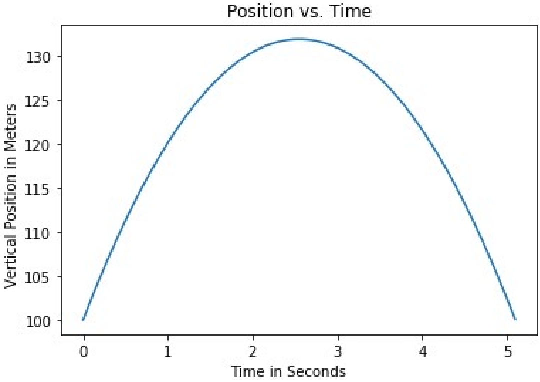
Our first goal with modeling 1-D motion is to get our code running so we can then progress to more complicated scenarios. Modifying equation (5) we can solve an equation for simple 1-D motion

$$\vec{a} = \frac{GM}{r^2} \tag{6}$$

where R is the radius and M is the mass the of Earth and $r = R + y$. The variable y is an arbitrary value that we choose off of initial conditions.

Simple 1-D Motion (4)

Using equation (6) where the initial conditions of our ball being thrown is $v_{iy} = 25\ m/s$ and $y_i = 100\ m$, we get the following plot when the motion is solved for.

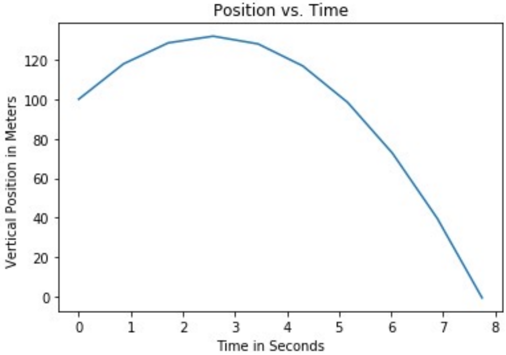


Simple 1-D Motion (5)

Neglecting air resistance we can find the time that it takes for our ball to reach the ground;

$$V_1 = V_0 + a\triangle t, \;\; V_1^2 = V_0^2 + 2a\triangle x, \;\; \triangle x = V_0\triangle t + \frac{1}{2}a\triangle t^2. \tag{7}$$

Using these three equations the time that it takes for the ball to reach the ground is $\triangle t = 7.74\,(s)$.



Simple 1-D Motion (6)

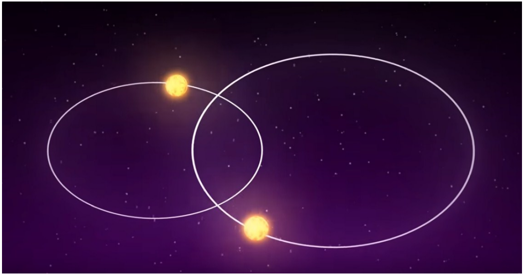
ODEint was used to solve the previous equations and plot their motion.

The next progression in this project was to plot 2-D kinematic motion, but because of time this part of the project was annexed and we moved on to 2-Body Motion.

Also, ODEint was not able to be used for a 2-D kinematic example and thus arose the necessity of developing my own code.

2-Body Motion (1)

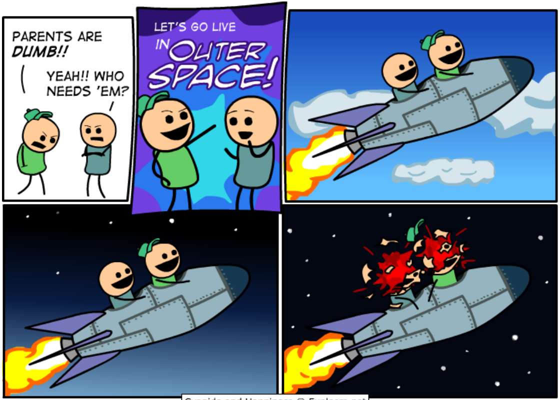
The purpose of modeling 1-D kinematic motion was to start with the easiest scenario and then progressively move towards more complicated scenarios.



The next more complicated scenario is the motion of two bodies in space.

2-Body Motion (2)

So let's begin with the simplest two body scenario, two bodies at rest.



Cyanide and Happiness © Explosm.net

2-Body Motion (3)

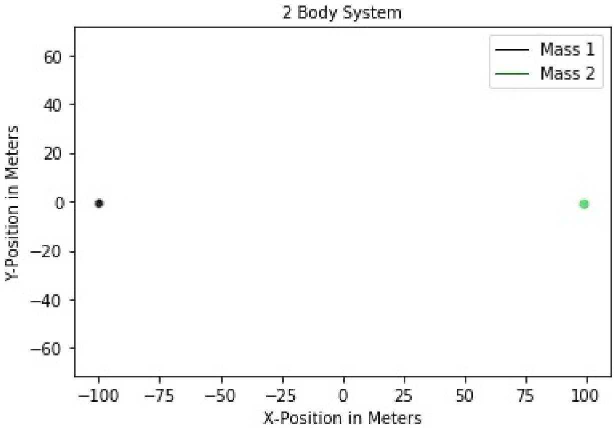
Here is an example of the 2-Body Code.

```
#################### Libraries
import matplotlib.pyplot as plt

#################### Constants
G=6.67e-11
M1=1.989*10**30
M2=6.39*10**23
AU=1.5e11
DS=24.0*60*60
GC=G*M1*M2

#################### Initital Conditions
##### Mass 1 Parameters
x1=0.0
y1=0.0
z1=0.0
xv1=0.0
yv1=15.0
zv1=0.0

##### Mass 2 Parameters
x2=1.524*AU
y2=0.0
z2=0.0
xv2=0.0
yv2=23996.0
zv2=0.0

#################### Time-Step
t=0.0
dt=1.0*DS

#################### Lists
x1list=[]
y1list=[]
x2list=[]
y2list=[]
```

```
#################### Loop
while t<1.86*365.25*DS: #Time interval for plots of motion
    ##### Compute the Force
    rx=x2-x1
    ry=y2-y1
    rz=z2-z1
    r3=(rx**2+ry**2+rz**2)**1.5
    fx=-GC*rx/r3
    fy=-GC*ry/r3
    fz=-GC*rz/r3
    ##### Update Quantities For 1
    xv1+=-fx*dt/M1
    yv1+=-fy*dt/M1
    zv1+=-fz*dt/M1
    x1+=xv1*dt
    y1+=yv1*dt
    z1+=zv1*dt
    ##### Update Quantities For 2
    xv2+=fx*dt/M2
    yv2+=fy*dt/M2
    zv2+=fz*dt/M2
    x2+=xv2*dt
    y2+=yv2*dt
    z2+=zv2*dt
    t+=dt
    ##### Save In Lists
    x1list.append(x1)
    y1list.append(y1)
    x2list.append(x2)
    y2list.append(y2)
```

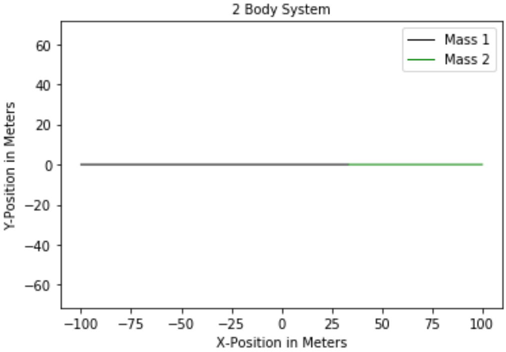
2-Body Motion (4)

Let's imagine two bodies of mass, $m_1 = 50\ kg$ and $m_2 = 100\ kg$ 200 meters apart originally.



2-Body Motion (5)

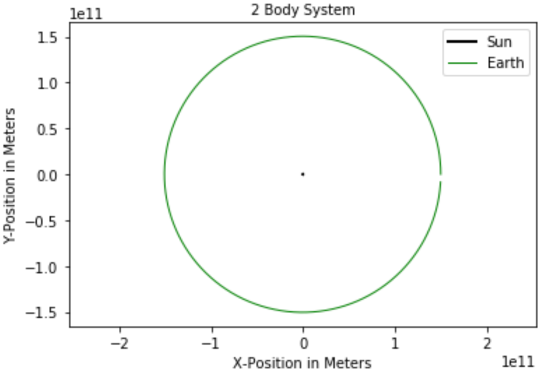
Intuitively, the bodies of mass should attract one another and eventually collide at a certain point. This journey takes ≈ 364 (Earth) days.



This confirms our code is working and we can move on to more complicated examples of 2 body motion.

2-Body Motion (6)

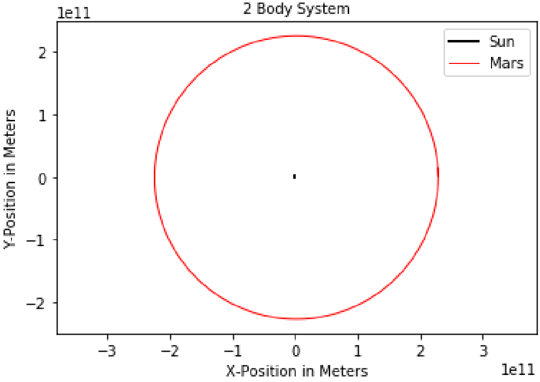
Now that our code is working on the simplest of scales, we can begin to plot orbits that are commonly known.



The code actually took $\approx$ 368 days to complete one orbit giving a 1% error in the code.

2-Body Motion (7)

And one more plot that is commonly known.



In this plot, it only took ≈ 672 days to complete one orbit yielding ≈ a 2% error.

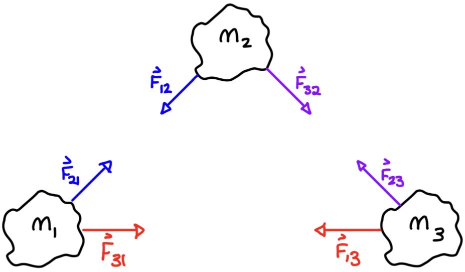
3-Body Motion (1)

The three body system is a very chaotic system, meaning the orbits are heavily dependent upon initial conditions.

The purpose of this part of the project was to probe initial conditions to determine what are suitable initial conditions for ternary systems.

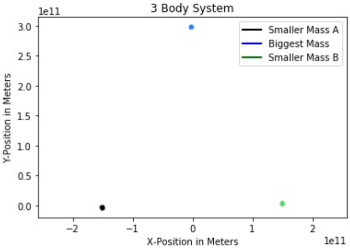
3-Body Motion (2)

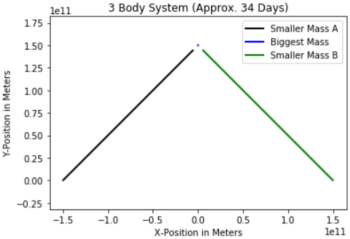
The first scenario that will be examined can be seen in the cartoon below.



3-Body Motion (3)

We will begin by examining a scenario of 3-bodies in space that are originally in an equilateral triangle orientation.

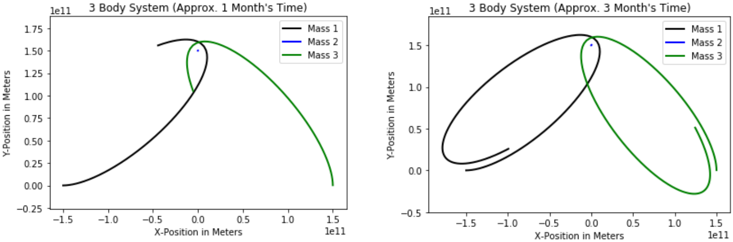




The black and green masses are roughly equivalent to the mass of Earth. The blue mass is roughly that of our sun.

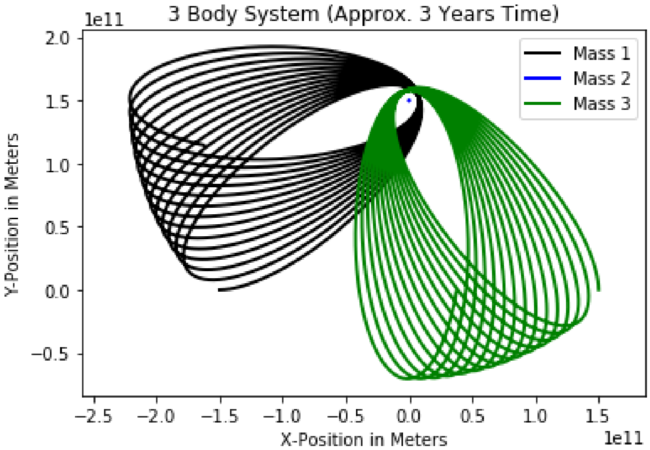
3-Body Motion (4)

Now we begin plotting these bodies in motion. M_1 will be moving at $\approx 30,000$ m/s in the x-direction and M_3 will be moving at $\approx 30,000$ m/s in the y-direction.



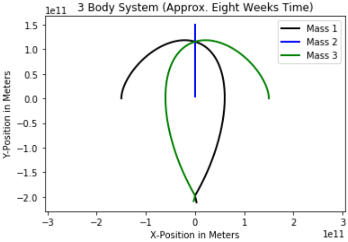
3-Body Motion (5)

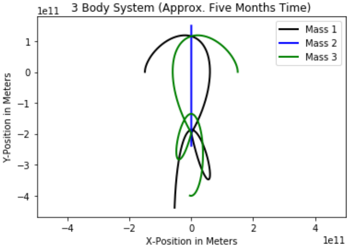
Letting this system progress over time, we get the following plot after about three years.



3-Body Motion (6)

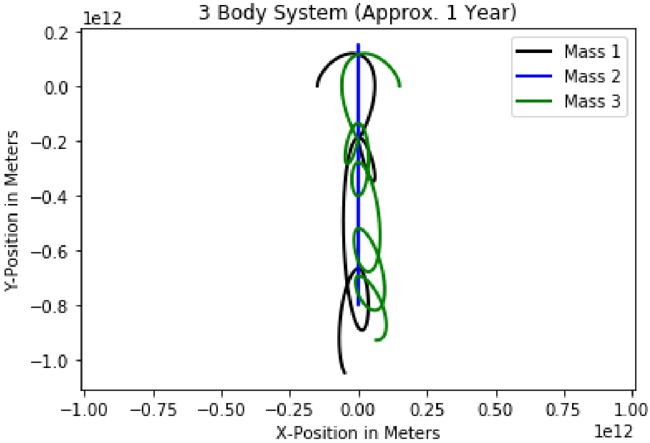
Using the same initial starting conditions as the last plots, with $M_1 = M_3 = 6 \cdot 10^{26}$ kg and both in motion at $\approx 30,000$ m/s and M_2 at $\approx -30,000$ m/s the following plots are created.





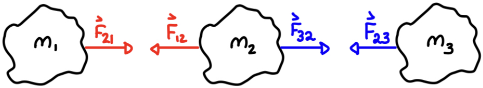
3-Body Motion (7)

Letting this system progress, the following final plot is created.



3-Body Motion (8)

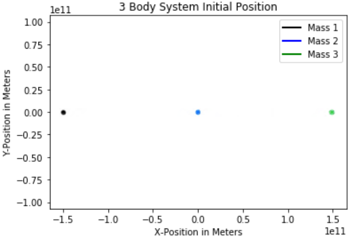
We now examine a new scenario where the three bodies are all aligned in a horizontal manner.

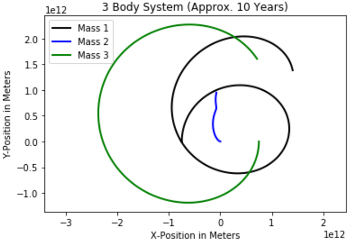


The masses of M_1 and M_3 are $\approx 6.0 \times 10^{29}$ kg.

3-Body Motion (9)

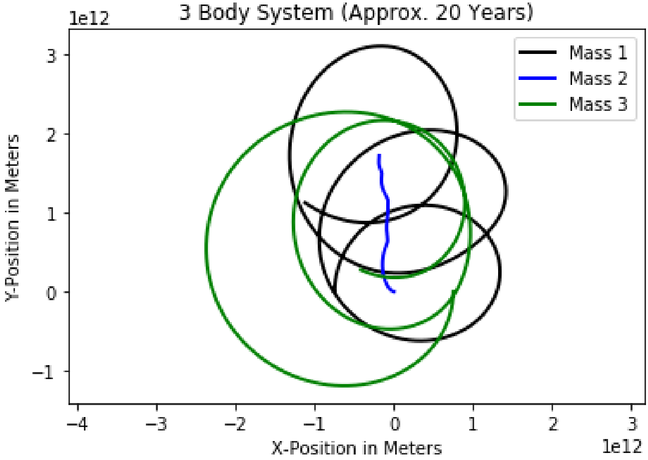
Mass 1 and Mass 3 are both initially moving at a velocity of $v = 50,000$ m/s, where Mass 1 is initially moving in the positive y-direction and Mass 3 is initially moving in the negative y-direction.





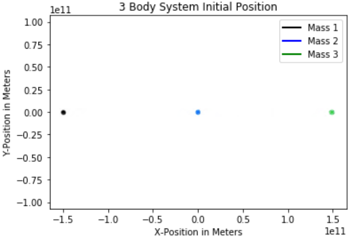
3-Body Motion (10)

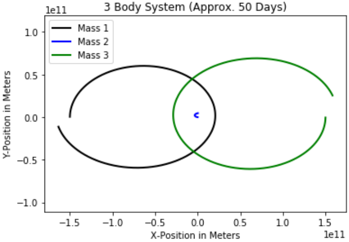
This system evolves over time and the following plot is generated after about 20 years time.



3-Body Motion (11)

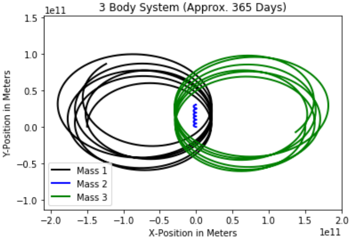
This scenario originates from the same initial positions as the last with M_1 and $M_3 \approx 2.0 \times 10^{29}$ kg. The initial velocities of M_1 and M_3 are still 30,000 m/s and -30,000 m/s in the y-direction respectively.

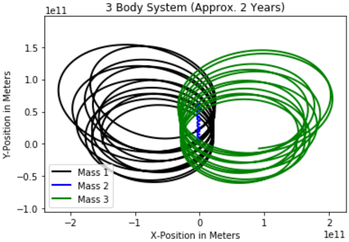




3-Body Motion (12)

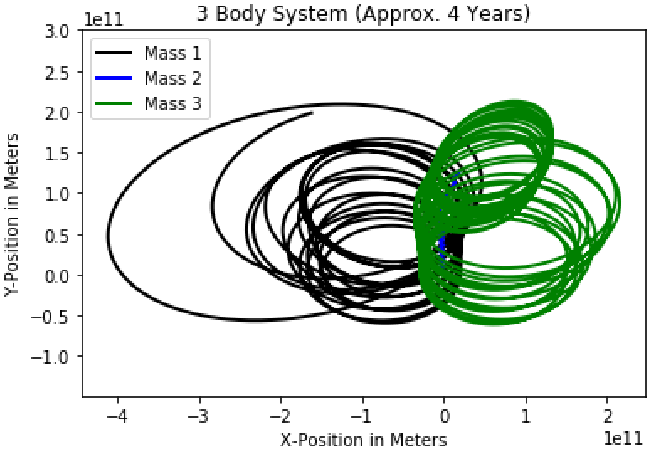
The purpose behind these plots are to depict smaller stars orbiting once another and show the non-predictable behavior of these systems.





3-Body Motion (13)

And lastly, the final plot of this system can be seen below.



3-Body Motion (14)

These plots show that certain systems are more unstable than others.

Larger mass ratios yield more stable ternary star systems.

Masses that are very similar in a system tend to yield more unstable systems that eventually devolve into binary star systems.

3-Body Motion (15)

The stability of these orbits comes into question, how can we determine if an orbit is going to be stable?

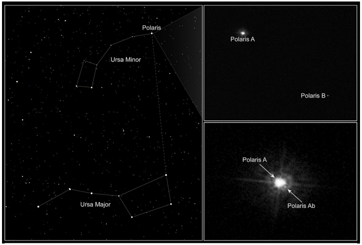
$$\frac{a_{out}}{a_{in}}|_{crit} = \frac{2.8}{1-e_{out}}\left(1-\frac{0.3i}{\pi}\right)\cdot\left(\frac{(1.0+q_{out})\cdot(1+e_{out})}{\sqrt{1-e_{out}}}\right)^{2/5} \tag{8}$$

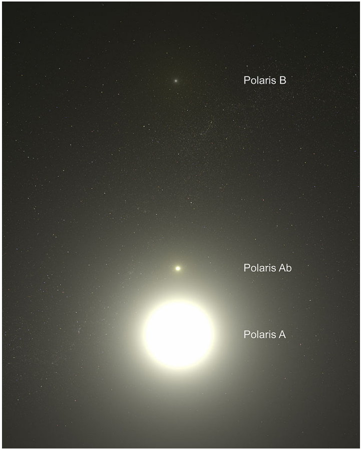
Orbits are considered unstable if $\frac{a_{out}}{a_{in}} < \frac{a_{out}}{a_{in}}|_{crit}$ where $i = \pi/2$ and $q_{out} = \frac{m_3}{m_1+m_2}$ [7]. a_{out} and a_{in} are the outer and inner semi major axis' of these orbits.

Stability of these orbits are largely determined on the eccentricity of the outer star in ternary star orbits.

3-Body Motion (16)

Here is an example of real three body orbit.





Conclusions

The three body systems are chaotic and usually have unpredictable orbits unless they are under ideal initial conditions.

There are an infinite amount of solutions for the three body system. Each solution arises from the initial conditions and thus is why we require numerical solutions.

In the search of stable ternary star systems we should be looking for large mass ratios and not very elliptic orbits.

References

Byrd, D. (2017, October 27). Perseus the Hero and a Demon Star. Retrieved September 29, 2019, from https://earthsky.org/tonight/close-up-on-constellation-perseus-the-hero.

(n.d.). How to Animate a Character Throwing a Ball. Retrieved from https://design.tutsplus.com/tutorials/how-to-animate-a-character-throwing-a-ball--cms-26207

CSIRO Australia Telescope National Facility, & Epping NSW. (2019, May 8). Introduction to Binary Stars. Retrieved September 29, 2019, from https://www.atnf.csiro.au/outreach/education/senior/astrophysics/binary_intro.html

(n.d.). Euler Forward Method. Retrieved from http://mathworld.wolfram.com/EulerForwardMethod.html

Perseus Constellation - Facts About Perseus. (n.d.). Retrieved from https://www.solarsystemquick.com/universe/perseus-constellation.htm.

Polaris. (2019, October 18). Retrieved from https://en.wikipedia.org/wiki/Polaris.

Toonen, S., & Zwart1, S. P. (2016, December 23). The evolution of hierarchical triple star-systems. Retrieved from https://comp-astrophys-cosmol.springeropen.com/articles/10.1186/s40668-016-0019-0.

Questions?

Any Questions?